

3. (Original) The method of claim 1 further comprising the step of:
providing an application shim for the application to receive the transformed event and provide the event to the application by using a native application program interface for the application.

4. (Original) The method of claim 3 further comprising the step of:
updating the application shim and the transformation profile responsive to changes in the application.

5. (Original) The method of claim 1 wherein the transformation profile includes a stylesheet.

6. (Original) The method of claim 1 wherein the transformation profile is stored in the directory.

7. (Currently amended) A software program for facilitating the use of a distributed directory running in a computer network, the program comprising being stored on a recordable medium and including instructions for:

receiving an first event from the distributed directory into an XML generator;
converting the first event into XML data;

transforming the XML data to a first predetermined format by a transformation processor, the first predetermined format being responsive to an application running in the computer network; and
transmitting the data transformed data according to the first predetermined format to the application;

receiving a third event from the distributed directory into an XML generator;

converting the third event into XML data;

transforming the XML data to a third predetermined format by the transformation processor, the third predetermined format being responsive to an application running in the computer network; and

transmitting the data transformed according to the third predetermined format to the application;

wherein the transformation processor includes an XSLT processor, the program further comprising instructions for:

providing a stylesheet to the XSLT processor, the stylesheet including formatting instructions for transforming XML data to the first predetermined format; and

providing a third stylesheet to the XSLT processor, the third stylesheet including formatting instructions for transforming XML data to the third predetermined format wherein the first stylesheet is different from the third stylesheet.

8. (Canceled)

9. (Currently amended) The software program of claim 8 7 further comprising instructions for:
receiving updates to the stylesheet responsive to any changes in either the distributed directory or the application.

10. (Original) The software program of claim 7 wherein the transformed data is transmitted to the application through an application shim to provide the transformed data to the application by using a native application program interface for the application.

11. (Original) The software program of claim 7 further comprising instructions for:
detecting the event through notification from an event handler of the distributed directory.

12. (Original) The software program of claim 7 further comprising instructions for:
receiving a second event from the application;
converting the second event into XML data;
transforming the XML data to a second predetermined format by the transformation processor,
the second predetermined format being responsive to the distributed directory; and
transmitting the data transformed according to the second predetermined format to the distributed directory.

13. (Currently amended) The software program of claim 12 wherein ~~the transformation processor includes an XSLT processor, the program further comprising~~ comprises instructions for:
~~providing a first stylesheet to the XSLT processor, the first stylesheet including formatting instructions for transforming XML data to the first predetermined format;~~
providing a second stylesheet to the XSLT processor, the second stylesheet including formatting instructions for transforming XML data to the second predetermined format.

14. (Original) The software program of claim 12 wherein the transformed data is transmitted from the application through an application shim.

15. (Currently amended) A software program for facilitating the use of a distributed directory running in a computer network, the program comprising instructions for:
receiving an event from ~~the~~ an application;
transforming the event to a predetermined format by a transformation processor, the predetermined format being responsive to the distributed directory; and
transmitting the transformed event to the distributed directory.

16. (Original) The software program of claim 15 further comprising instructions for:
converting the event into markup language data prior to transforming the event.

17. (Original) The software program of claim 15 further comprising instructions for:
providing a transformation profile to the transformation processor, the transformation profile including formatting instructions for transforming the markup language data to the predetermined format.

18. (Original) A distributed computer system comprising:
a first processor connected to a network for executing computer code;
a second processor connected to the network for executing computer code;
a first memory connected to the first processor;
a second memory connected to the second processor;
a distributed directory, a portion of which being stored in the first memory;
an application, a portion of which being stored in the second memory;
a transformation profile for defining a predetermined format for use by the application;
software for detecting an event in the distributed directory;
software for transforming the event to the predetermined format by using a generic transformation tool and the transformation profile; and
software for providing the transformed event to the application;
whereby the application becomes aware of the event by having the event provided to the application in a transformed state.

19. (Original) The system of claim 18 further comprising:
software for converting the event to a generic data description before transforming the event.

20. (Original) The system of claim 18 further comprising:
an application shim for the application to receive the transformed event and provide the event to
the application by using a native application program interface for the application.

21. (Original) The system of claim 18 further comprising:
a second transformation profile for defining a second predetermined format for use by the
distributed directory;
software for transforming an application event to the second predetermined format by using the
generic transformation tool and the second transformation profile; and
software for providing the transformed application event to the distributed directory;
whereby the distributed directory becomes aware of the application event by having the application event
provided to the distributed directory in a transformed state.

22. (Original) The system of claim 21 wherein the generic transformation tool utilizes a
markup language and the software for transforming the event and the application event utilizes a
transformation processor.